

物体検出とロボットアームを用いたアプリ稼働監視システムの開発

井崎俊太郎¹ 堀川三好¹ 岡本東¹

概要: 近年、働き方改革が進められる中、定型業務に対してソフトウェアロボットを用いて自動化する RPA (Robotic Process Automation) が、オフィス分野の効率化として注目を集めている。本研究では、ロボットアームと物体検出を組み合わせることで、複雑な定型業務や非定型業務を自動化する RPA の実現を目的としている。本稿では、スマートフォンのアプリケーション稼働監視を自動化するために必要なシステム構築および物体検出のための効率的なデータセット収集方法について報告する。

キーワード: ロボットアーム 物体検出 機械学習 アプリ稼働監視

1. はじめに

生産労働人口の減少や働き方改革と合わせて、Robotic Process Automation (RPA) がオフィス分野の効率化において注目を集めている [1]。RPA は、これまで人間が行ってきた PC やサーバ上の定型業務に対して、ソフトウェアロボットを用いることで自動化する技術である。また、Enhanced Process Automation (EPA) や Cognitive Automation (CA) に向けて、画像認識や物体検出の AI 技術と組み合わせることで、非定型業務に対応した RPA の事例が見られる。物体検出は、入力画像に対して画像処理や機械学習を適用することで特定の物体を発見する AI 技術の一つである。RPA と同様に小売業界や医療業界などの幅広い分野で活用されており、日々新しい手法が提案されている。

一方で、従来の RPA は、ソフトウェアのみで情報システムの自動化に取り組む事例が多く、軽微な仕様変更に伴ってシステム改善やパラメータの調整が必要となる課題がある。また、自動化できる範囲が情報システム内に限定されるため定型業務を中心とした事例が多く、非定型業務への適用範囲拡張も課題となる。前者の課題は、AI 技術を用いることで、ある程度の仕様変更に対応できる柔軟な RPA の実現が見込めるものの、画像認識や物体検出技術をどのように適用するかが重要となる。特に、機械学習を用いるには、効率的なデータセット収集方法を明らかにする必要がある。後者の課題は、近年のロボティクス技術の進展の中で、安価かつ高機能なロボットアームが利用可能となり、人間の腕の動作の代替しつつ、ソフトウェア的な自動化をはかる RPA の適用範囲の拡張が期待できる。

本研究は、ロボットアームと物体検出を組み合わせることで人間の腕の動作を再現し、オフィス業務での複雑な定型業務や非定型業務を自動化する新しい RPA の実現を目的としている。本稿では、企業と共同研究で開発したスマートフォン用アプリケーション (以後、アプリと呼ぶ) の稼働監視業務を自動化するシステムについて報告する。アプリでは、広告配置や機能の追加に伴いアイコン等のレイアウト変更が頻繁に行われる。そのため、提案システムで

は、ある程度のレイアウト変更に対応できる必要がある。また、デザインの一新が定期的に行われるため、効率的かつ簡易にシステム調整や導入を行える必要がある。本稿では、これらを考慮して開発した提案システムの概要について報告する。また、物体検出に利用する学習用データセット (以後、データセット) の構築に必要な、画像データ収集方法の比較実験について報告をする。

2. 関連研究・事例

RPA に関しては、企業での多くの導入事例が報告されている。例えば、NTT データでは WinActor と呼ばれる RPA ツールを開発しており、システムの運用ログを管理・集計する業務に活用して人間による最終確認以外の業務を自動化し、大幅な作業効率化に成功している [2]。また、三井住友海上火災保険では社員の PC 操作の調査を行い、全体の PC 操作の内、18%の操作を RPA によって自動化可能であることを報告している [3]。RPA の適用事例については、オフィス業務での PC 操作やサーバ内部処理の自動化を対象とすることが多い。ロボティクス技術を RPA に活用する取り組みは、ものづくり現場での導入に比べると活用事例は少ないが、自動で押印をするロボット [4] 等の一部事例が見られ始めている。

また、AI 技術によるシステム異常検知に関する研究は、センサー値、システム稼働ログおよび画像を用いた 1-class SVM [5] や k 近傍法 [6] 等により、正常状態から離れた値を検出する研究が多い。多くの先行研究では、CIFAR-10 [7]等のオープンデータや企業が蓄積している大規模データを用いた検証実験が多い。小規模データに活用できる研究 [8] [9] も見られるが、適用事例に応じたデータセットの構築方法に関する研究は少ない。特に、画像データの収集においては、対象とする業務や環境に依存することが多いため、適用事例に応じた収集方法の確立が必要となる。先行研究 [10] では、オフィス業務で物体検出を適用する際に課題となる照度変化に適した画像データ収集方法について比較実験を行った。その結果、照度変化に強いデータセット構築方法を明らかにしたが、より簡易な学習モデル

¹ 岩手県立大学大学院
Iwate Prefectural University Graduate school

の構築方法および画像データ収集方法を確立することが課題となっていた。

3. アプリ稼働監視システム

3.1 システム概要

単一システムで稼働するアプリの場合は、システムログからの稼働状態の確認は容易である。しかしながら、近年のアプリは、外部の複数企業にまたがるプラットフォームやサーバを経由して、利用者にコンテンツを提供する形態が多い。また、頻繁な仕様変更に伴う開発バグやサイバー攻撃等からアプリの稼働状態を常に見守る必要がある。そのため、アプリを提供する企業では、システムログの監視と併せて社内の人間が定期的に操作を行い、稼働状態の確認を行う事例が見られる。これにより、最終的な利用者の立場からアプリの稼働状態を確認することが可能となるが、運用負荷としては大きなものとなる。

提案するアプリ稼働監視システムでは、この人間が定期的に行う確認操作を自動化する。すなわち、物体検出を用いてアプリのボタンの位置を検出し、ロボットアームでタップする。その結果のアプリ画面から稼働状態を判定し、稼働状態をアプリ運営者に通知する仕組みである。これにより、具体的な原因分析は出来ないまでも、システムの異常にいち早く気づき、利用者へのサービス復旧に寄与する仕組みを構築する。

3.2 システム要件

提案システムを開発するにあたっては、共同研究先企業からの以下のような要望があった。

(1) 低価格・容易な実務導入

産業用ロボットアームや画像認識を用いての導入は容易であるが、高価格になるため避けたい。また、新規アプリの導入時のみでなく、仕様変更や簡易なレイアウト変更が頻繁に行われることから、運用準備および容易な運用が行える仕組みにしたい。

(2) オフィス環境での利用

オフィス環境では、照明器具や天候の影響を受けやすく、画像処理において照度変化の影響が大きいため考慮する必要がある。また、頻繁に人間の動きが生じるため、若干の位置ずれや振動に柔軟な仕組みを実現したい。

(3) 複数アプリの監視

監視間隔は対象アプリにより異なるが、導入コストの観点から複数アプリを1つのシステムで監視可能としたい。

これらの要望に応えるため、教育および軽産業用の安価なロボットアームを利用するのに併せて、クラウド上で専門知識がなくても容易に物体検出の学習モデルを作成できるサービスを取り入れてシステム開発することとした。また、定常的なアプリ稼働監視であれば、テンプレートマッチングを用いた稼働状態の判定も技術的には十分に可能である

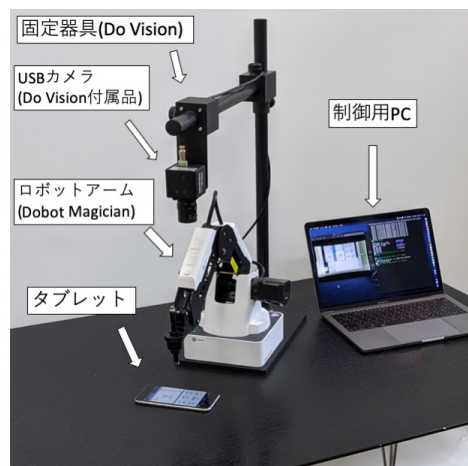


図 1. ハードウェア構成

が、オフィス環境での柔軟な運用、仕様変更の頻度や導入準備の容易性から物体検出技術を用いることとした。

3.3 ハードウェア構成

提案システムのハードウェア構成を図 1 に示す。ロボットアームは、教育用として開発され広く利用されている DOBOT Magician (DOBOT 社製、約 15 万円) を用いる。アームの先端部分に一般的に使われているタブレット操作作用のタッチペンを装着し、タブレット (iPad Air) の操作を行う。また、カメラとして DO Vision (DOBOT 社製、約 8 万円) を固定し、物体検出に用いる画像を取得する。ロボットアーム制御用および物体検出処理用の制御 PC (以後、制御 PC) は、市販のノート PC (約 20 万円) を用いる。

3.4 ソフトウェア構成

開発システムのソフトウェア構成を表 1 に示す。先行研究 [10] では、物体検出に YOLO ver.3 を使用したが、データセット収集や学習モデルの構築に専門的知識が必要であった。そのため、提案システムでは実務導入の容易性を考慮し、Google AutoML Vision (以後、AutoML) を使用する。AutoML は、Neural Architecture Search [7] 等の Auto Machine Learning や転移学習の技術を用いて、利用者が用意したデータセットから高精度の学習モデルを作成することができるクラウドサービスである。

開発システムでは、AutoML で作成した学習モデルをダウンロードし、制御 PC にインストールした TensorFlow Lite を用いて物体検出する。OpenCV を用いてカメラから画像

表 1. ソフトウェア構成

OS	Windows10
プログラミング言語	Python3.7
計算処理	Numpy
画像取得・加工	OpenCV3 Pillow
物体位置推論	TensorflowLite
学習モデル作成	Google AutoML Vision Edge

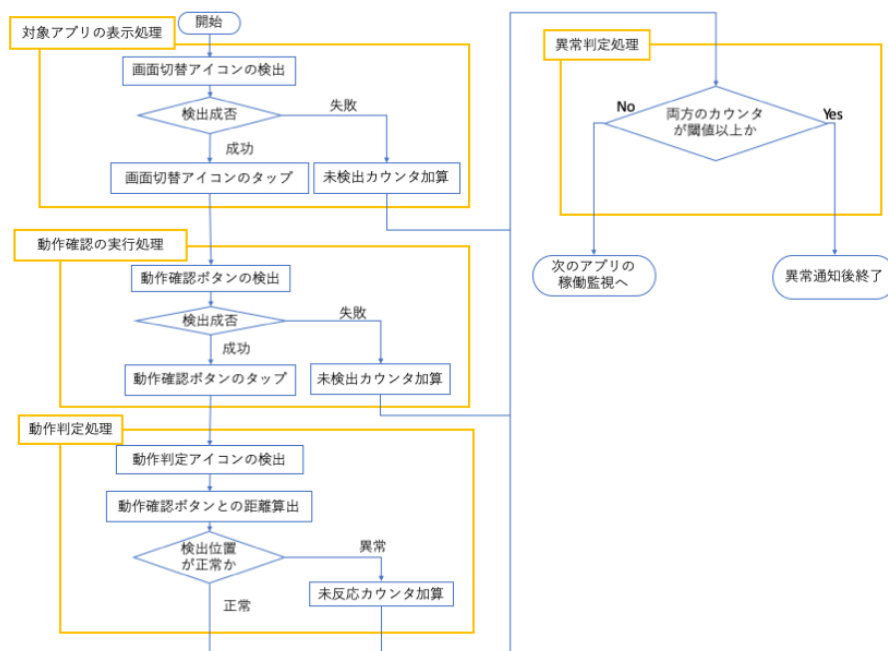


図 2. 処理フロー

を取得し、Pillow を使用して画像を Tensorflow Lite で扱う形式に変換する。ロボットアームの制御には、DOBOT 社から提供されるライブラリを用いて開発した Python プログラムを制御 PC 上で実行する。

3.5 提案システムの処理の流れ

提案システムの処理フローを図 2 に示す。提案システムは、対象アプリの表示、動作確認の実行、動作判定および異常判定の 4 つ処理で構成される。事例として 2020 年 8 月時点の LINE アプリを取り上げ、

(事例 1) JR 東日本チャットボット (図 3)

(事例 2) 東北電力チャットボット (図 4)

を例に各構成部品の説明および処理を説明する。

(1) 対象アプリの表示処理

複数アプリの同時監視を行うため、画面切り替えアイコンをロボットアームでタップして、対象アプリを表示する。

事例では、画面切り替えアイコンとして LINE アプリのプロフィールアイコンを検出し、タップすることでチャットルームにおいて事例 1 と事例 2 のチャットルームを切り替えながら交互に監視を行う。

(2) 動作確認の実行処理

各アプリに定めた動作確認ボタンの物体検出を行い、ロボットアームでタップする。物体検出に失敗した場合は、アプリ表示がされていないと判断し、未検出カウンタを加算して異常判定処理に進む。

事例 1 においては、下部に複数のメニューボタンが表示されている。この中から、運行情報を表示させるボタンを動作確認ボタンとして物体検出し、タップすることでアプリの動作結果を表示する。



図 3. アプリ構成部品の例 (事例 1)



図 4. アプリ構成部品の例 (事例 2)

(3) 動作判定処理

動作結果の画面から、各アプリに定めた動作判定アイコンの物体検出を行う。その後、動作確認ボタンと動作判定アイコンの位置関係から、アプリの動作結果が正しいかの判定を行う。位置関係が正しい場合は、監視対象アプリを切り替えて対象アプリの表示処理に戻る。位置関係が正しくない場合は、未反応と判定を行い動作確認ボタンのタップを一定回数繰り返す。一定回数繰り返しても未反応と判定された場合は、未反応カウンタを加算して異常判定処理へ進む。

事例 1 においては、動作確認ボタンをタップした際に表示される運行情報のヘッダー部分を動作判定アイコンとし、動作確認ボタンとの距離から動作判定をする。また、未反応時の繰り返し数を 3 回と設定する。図 5 に正常時の画面を示す。図 6 にネットワーク障害等によりアプリの稼働状態が異常な場合に、動作確認ボタンを複数回タップした事例を示す。正常時では運行情報が正しい位置に表示され、動作判定アイコンと動作確認ボタンの距離が一定となる。しかし、異常時では反応では運行状況が表示されず、距離が長くなる。

(4) 異常判定処理

未検出カウンタと未反応カウンタの値を確認し、閾値以上であるかを確認する。閾値以上の場合は異常通知を送信する。また、閾値以下の場合には警告通知を記録し、監視対象アプリを切り替えて対象アプリの表示処理へ戻る。

試験運用において、一回の未反応や未検出で稼働状態を判断した場合、正常に稼働しているにも関わらず異常通知を送信する事例が見られた。例えば、タブレットのネットワーク環境が不安定で瞬間的にアプリの反応が遅れている場合等である。そのため、未検出および未反応のカウンタを準備し、複数回連続した異常を確認した上で異常通知を送信することとした。

事例では、各カウンタの回数を 5 回と設定する。事例 1 と事例 2 のアプリ監視は、約 3 分で 1 周するように調整しているため、異常発生時には約 7 分以内で異常通知を送信することとなる。また、異常および警告通知については、Microsoft Teams 上にシステム監視ログ用のチームを作成し、API 経由で記録することとする。

4. データセット収集方法の検証

4.1 実験概要

一般的に、物体検出で高精度な検出を行うには、数千枚から数万枚程度と大量の画像データのデータセットが必要になる。しかしながら、アプリのような仕様変更が頻繁に起こる場合には、簡易な学習モデルの構築方法と効果的なデータセット収集方法を検討する必要がある。提案システムで用いている AutoML では、ユースケースに合わせて学習モデルを設計し、最適化された学習モデルではないもの



図 5. 正常時の画面

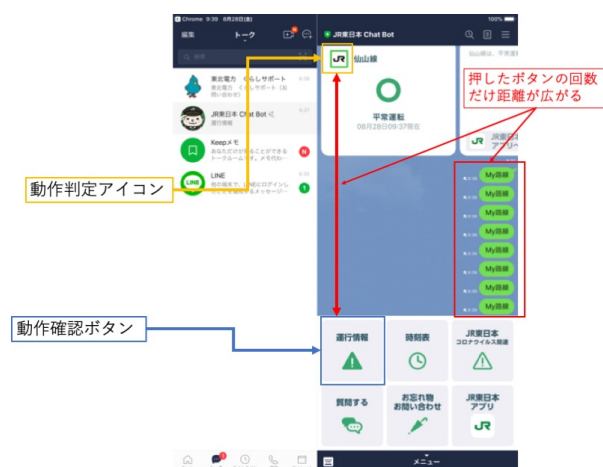


図 6. 異常発生時の画面

の実用性があり、比較的少ない画像枚数でも十分な精度を得られる学習モデルの作成が可能である。しかしながら、効率的なデータセットの構築方法は、利用環境によって異なるため、今回の事例についてデータセット収集方法の比較検証を行い、実務運用への知見を得る。

4.2 実験方法

学習モデル構築のための AutoML の設定は、データセット中 7 割の画像を訓練用、3 割を検証用として、精度重視の設定で学習を行う。また、物体検出精度の評価指標として、mean Average Precision (以後、mAP) を用いる。mAP は、PASCAL Visual Object Classes Challenge 2012 [11] で用いられた評価指標で、真陰性を評価に含めず未検出と誤検出を同時に評価することができ、物体検出の精度検証で幅広く使われている。平均適合率 (AP) は、物体検出において正解領域と予測領域の一致比率を示す IoU (Intersection over Union) をもとに正解・不正解を判定した場合に、適合率-再現率曲線から算出される。mAP は平均適合率の平均となる。本実験では、IoU を 0.5 とする。評価に使用する画像は、事例 1 において 300 枚のランダムに撮影した画像を用いる。例えば、台の上で水平に回転したものや、手で持

ち上げて傾きをつけた画像が含まれている。実験は、まず、画像収集方法による影響を検証した後、さらに画像枚数の影響を明らかにする。

4.3 画像収集方法による影響

画像収集方法により、物体検出精度にどのような影響があるかを検証する。収集方法として、回転角度として 90 度と 45 度、傾斜角度として 0 度（水平）と 30 度を組み合わせた 4 方法でデータセット収集を比較する。AutoML では、1 つの正解ラベルに 10 枚以上の画像が必要となるため、収集枚数を調整する。収集した画像例を図 7 に示す。

(1) データセット 1（回転角 90 度・傾斜角 0 度）

タブレットを水平に設置し、90 度ずつ回転させて撮影した画像でデータセットを構築する。1 回転あたり 4 枚の画像が取れるため 3 回転分の画像を収集して 12 枚の画像で構成する。

(2) データセット 2（回転角 45 度・傾斜角 0 度）

タブレットを水平に設置し、45 度ずつ回転させて撮影した画像でデータセットを構築する。1 回転あたり 8 枚の画像が取れるため 2 回転分の画像を収集して 16 枚の画像で構成する。

(3) データセット 3（回転角 90 度・傾斜角 30 度）

タブレットを 30 度の傾斜ができるように持ち上げ、90 度ずつ回転させて 4 方向から撮影した画像でデータセットを構築する。1 回転あたり 4 枚×4 方向の 16 枚の画像が撮れ、傾斜を付けない 4 枚の画像を加え 20 枚の画像で構成する。

(4) データセット 4（回転角 45 度・傾斜角 30 度）

タブレットを 30 度の傾斜ができるように持ち上げ、45 度ずつ回転させて 4 方向から撮影した画像でデータセットを構築する。1 回転あたり 8 枚×4 方向の 32 枚の画像が撮れ、傾斜を付けない 8 枚を加えて 40 枚の画像で構成する。

実験結果を表 2 に示す。どのデータセットを用いた学習

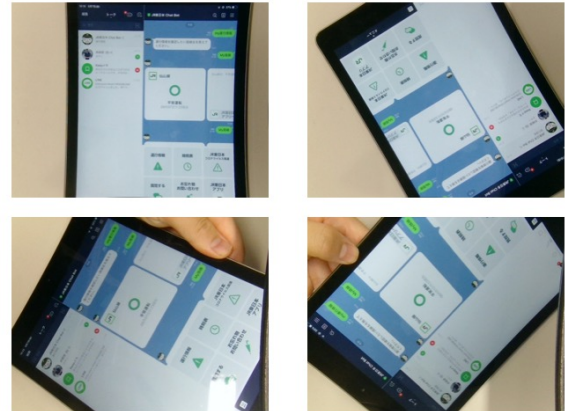


図 7. データセット構築に使用した画像例

モデルにおいても、画面切り替えアイコンの平均適合率が高い。一方で、動作確認ボタンや動作判定アイコンは、データセット 1 および 3 での平均適合率が低いため、mAP も低くなる。事例 1 のような特徴的な画面切り替えボタンの場合は、どのようなデータセットでも物体検出することが可能だが、動作確認ボタンや動作判定アイコンのような比較的単純な形状の場合には、特に回転角 45 度付近の物体検出に失敗した。すなわち、学習データが不足していると考察できる。また、傾斜角度については mAP への影響は小さい。そのため、少ない画像枚数であり、mAP が高いデータセット 2 が適切なものと考察できる。

4.4 画像枚数による影響

データセット 2 の収集方法に対して、画像枚数による影響を明らかにする。データセット 2 の 16 枚、画像枚数が 24 枚になる 3 回転、および画像枚数が 104 枚以上になる 13 回転して撮影したデータセットで比較する。

実験結果を表 3 に示す。16 枚と 24 枚の場合には精度が大きく変わらないが、104 枚の場合には精度が下がる。画像収集方法が同じであるため、104 枚の場合は同じような画

表 2. 画像収集方法による影響（AP と mAP）

	データセット 1	データセット 2	データセット 3	データセット 4
画面切り替えアイコン	0.985	0.983	0.778	0.980
動作確認ボタン	0.530	0.993	0.525	0.990
動作判定アイコン	0.253	0.895	0.179	0.820
mAP	0.589	0.957	0.494	0.930

表 3. 画像枚数の影響（AP と mAP）

	16 枚	24 枚	104 枚
画面切り替えアイコン	0.983	0.994	0.997
動作確認ボタン	0.993	0.996	0.996
動作判定アイコン	0.895	0.851	0.763
mAP	0.957	0.947	0.919

表 4. 画像収集方法の追加実験 (AP および mAP)

	データセット 5	データセット 2	データセット 6
画面切り替えアイコン	0.988	0.983	0.996
動作確認ボタン	0.639	0.993	0.996
動作判定アイコン	0.273	0.895	0.888
mAP	0.633	0.957	0.960

像が発生し、過学習を起こしたと考察される。そのため、回転角 45 度のデータ収集の場合は、16 枚程度にするのが適切であると考察できる。

4.5 画像収集方法の追加実験

ここまでの実験で、データセット 2 の収集方法、すなわち、回転角 45 度かつ傾斜角 0 度での画像データの収集で十分な物体検出精度を得られた。より詳細な回転角度での影響を明らかにするため、追加実験を行う。

(1) データセット 5 (傾斜角 0 度・回転角 90 度)

データセット 1 と同様であるが 4 回転させて 16 枚の画像を収集する。

(2) データセット 6 (傾斜角 0 度・回転角 22.5 度)

タブレットを水平に設置し、22.5 度ずつ回転させて撮影した画像でデータセットを構築する。1 回転分の画像を収集して 16 枚の画像で構成されている。

実験結果を表 4 に示す。画面切り替えアイコンにおいては、どのデータセットも同様に高い平均適合率を示している。一方で、動作確認ボタンと動作判定アイコンの平均適合率が低いため、データセット 5 の mAP が低い。また、データセット 2 と 6 の mAP の差は小さい。従って、今回の事例では、傾斜角 0 度、すなわち水平にタブレット置き、22.5 度または 45 度程度の回転角度で 16 枚程度の写真を撮影し、データセットにすることが適切だと考察できる。

5. おわりに

本研究では、スマートフォン用アプリケーションの稼働監視業務を自動化するシステムを提案し、開発したシステムについて報告した。また、物体検出を行う学習モデルに必要なデータセットの収集方法について比較し、実務導入の際に必要なタブレット画面の撮影方法や必要枚数を考察した。開発したシステムは、共同研究先企業で事業化を行い、サービス提供を開始している。今後は、システムの信頼性の向上や今回の事例以外のアプリでの導入に取り組む予定である。

画像認識や物体検出は、既に身近な技術として実務導入する段階であり、少ない専門知識でも容易にシステム適用が可能である。本研究は、オフィス業務での複雑な定型業務や非定型業務を自動化する新しい RPA の実現を目的として取り組んでおり、他のオフィス業務の自動化も検討する予定である。また、今後は、これらの技術が未適用の分

野・業種での導入支援やノウハウの蓄積にも取り組んで行きたい。

参考文献

- [1]RPA(働き方改革:業務自動化による生産性向上), 総務省, https://www.soumu.go.jp/menu_news/s-news/02tsushin02_04000043.html (参照 2020 年 3 月 17 日)
- [2]研究開発リーダー, 中川 拓, Vol.15, No.8 2018
- [3]西村 崇, 玉置亮太「本番 RPA」, 「日経コンピュータ」, 2017 年 11 月 23 日, p22-39
- [4]日刊工業新聞「国際ロボット展、開幕 東京ビッグサイトで 21 日まで」2019 年 12 月 19 日
- [5]Support Vector Method for Novelty Detection, Bernhard Scholkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, John Platt, Advances in Neural Information Processing Systems 12(1999)
- [6]A Clustering Method Based on K-Means Algorithm, YouguoLi, HaiyanWu, Physics Procedia, Volume 25, Pages 1104-1109(2012)
- [7]CIFAR-10 and CIFAR-100 datasets, Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, <https://www.cs.toronto.edu/~kriz/cifar.html> (参照 2020 年 6 月 19 日)
- [8]Siamese Neural Networks for One-shot Image Recognition, Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, ICML Deep Learning Workshop. Vol. 2(2015)
- [9]RandAugment: Practical automated data augmentation with a reduced search space, Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, Quoc V. Le, arXiv:1909.1371(2019)
- [10]ロボットアームと物体検出を用いたスマホアプリ監視の自動化, 井崎 俊太郎, 堀川 三好, 岡本 東, 第 82 回全国大会講演論文集, 239-240(2020)
- [11]ThePascalVisualObjectClassesChallenge2012(VOC2012) Development Kit, Mark Everingham, John Winn, http://host.robots.ox.ac.uk/pascal/VOC/voc2012/devkit_doc.pdf (参照 2020 年 6 月 19 日)